Project 2: Simple FTP with Go-back-N

Aparna Patil - akpatil
Nikunj Shah - nmshah5

In this project we have implemented simple FTP protocol using UDP to send packets and for reliable delivery we have used Go-back-N scheme. We have carried out 3 experiments to evaluate the effect of the window size N, MSS, and packet loss probability p on the total delay for transferring a large file. For the experiments we were separated by 7 hops on our own machine and machine on university network and RTT between the two hosts is 2.79 ms. For transmission we are using file size of 1.1 MB.

Task 1: Effect of window sixe N

For Task 1 we have considered MSS as 500 bytes, loss probability as p=0.05 and file size greater than 1.1 MB for transfer. We have run the Go-back-N protocol to transfer the file selected by varying value of the window size as N = 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024. We have transmitted the file 5 times for each value of N.

Following table represents data obtained for various iterations for value of N:

Task 1										
Server	Client	N	MSS	Probability	RTT1	RTT2	RTT3	RTT4	RTT5	Avg. RTT
EOS Machine	Laptop	1	500	0.05	26.97	26.14	27.57	26.62	26.37	26.73
		2	500	0.05	20.1	22.18	19.49	22.50	20.64	20.98
		4	500	0.05	17.3	25.14	18.11	18.85	21.11	20.10
		8	500	0.05	16.4	16.98	17.75	17.33	18.10	17.32
		16	500	0.05	15.7	17.46	17.28	17.90	17.63	17.20
		32	500	0.05	18.6	19.31	17.54	17.96	18.98	18.48
		64	500	0.05	17.1	18.36	18.14	18.95	19.45	18.40
		128	500	0.05	23.0	23.44	21.77	22.33	25.95	23.30
		256	500	0.05	32.8	33.93	28.94	32.37	34.61	32.54
		512	500	0.05	39.8	34.33	38.31	35.13	35.95	36.70
		1024	500	0.05	45.2	53.36	51.04	38.57	46.20	46.87

The graph in figure 1 below plots the average round trip transfer time against N. In this graph, values of window size are plotted on the x-axis and values obtained for average round trip time are plotted on y-axis.

Project 2: Simple FTP with Go-back-N

Aparna Patil - akpatil
Nikunj Shah - nmshah5

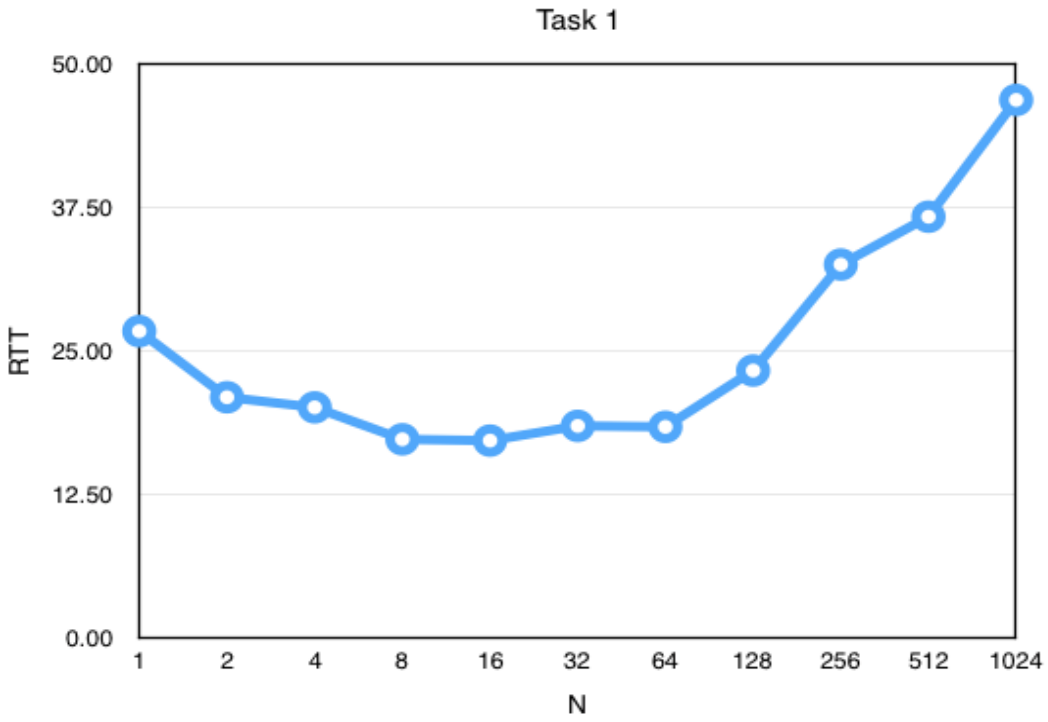

Figure 1: Transfer Time (seconds) vs. Window Size (N)

In the Go-back-N protocol, if the window size is small then receiver will consume the packets very slowly leading to multiple timeouts at sending end since ACKs will be received very slowly at sending end. On the other hand, with large window size if packet is lost due to timeout or ACK loss then all the packets in the non-acknowledged packets in the window are sent leads to higher transfer time. Hence, ideally best window size should be for values of N in middle range. According to our results best window size is for N=64.

Task 2: Effect of MSS

In this experiment, we have considered window size N = 64 and the loss probability p = 0.05. Go-back-N protocol to transfer the same file was run with varying the MSS values from 100 bytes to 1000 bytes in increments of 100 bytes. For each value of MSS, we have transmitted the file 5 times, and computed the average delay over the five transmissions.

Following table represents values obtained for varying MSS values:

Project 2: Simple FTP with Go-back-N

Aparna Patil - akpatil
Nikunj Shah - nmshah5

Task 2										
Server	Client	N	Probability	MSS	RTT1	RTT2	RTT3	RTT4	RTT5	Avg. RTT
EOS Machine	Laptop	64	0.05	100	96.3	105.7	112.0	100.9	103.9	103.7
		64	0.05	200	48.2	50.9	46.1	41.4	51.4	47.6
		64	0.05	300	27.7	29.4	34.5	31.0	29.1	30.3
		64	0.05	400	21.5	23.7	23.4	26.6	22.7	23.6
		64	0.05	500	18.27	21.89	21.21	22.07	20.75	20.84
		64	0.05	600	17.19	15.77	21.13	16.67	16.02	17.35
		64	0.05	700	15.08	18.58	14.64	16.37	14.57	15.85
		64	0.05	800	12.25	14.03	13.94	12.80	12.58	13.12
		64	0.05	900	11.66	13.53	12.19	10.81	12.27	12.09
		64	0.05	1000	12.07	12.31	9.77	12.62	8.55	11.06

The graph in figure 2 below plots the average round trip transfer time against MSS. In this graph, values of MSS are plotted on the x-axis and values obtained for average round trip time are plotted on y-axis.

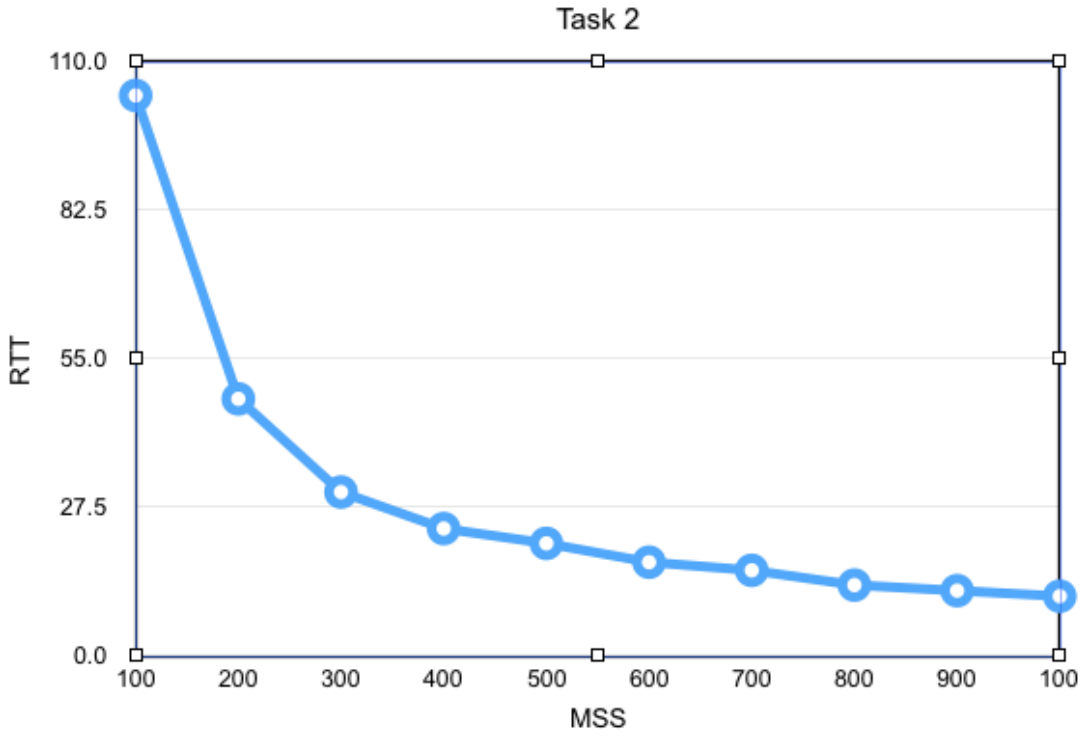

Figure 2: Transfer Time (seconds) vs. Maximum Segment Size (MSS)

In graph plotted above is exponentially decreasing as MSS value increases. This behavior is expected. We know in Go-back-N, as MSS is increased, the average delay decreases exponentially and it is dependent on the bandwidth of the network. A larger MSS works best only if network bandwidth can support it efficiently. Low network bandwidth cannot handle larger size segment, also smaller segment size in a network with a large bandwidth would be an inefficient utilization of bandwidth resulting in a higher transfer time relative to a larger segment size.

Task 3: Effect of loss probability p

For this experiment, we have considered the MSS as 500 bytes and the window size N = 64. We have executed the Go-back-N protocol to transfer the same file by varying the loss probability from p = 0.01 to p = 0.10 in increments of 0.01. For each value of p we have transmitted the file 5 times, and computed the average delay over these five transfers.

Following table represents values obtained for varying probability values:

Task 3										
Server	Client	N	MSS	Probability	RTT1	RTT2	RTT3	RTT4	RTT5	Avg. RTT
EOS Machine	Laptop	64	500	0.01	8.4	9.4	8.4	8.0	7.7	8.4
		64	500	0.02	12.0	12.0	10.8	11.6	7.0	10.7
		64	500	0.03	14.2	14.1	16.0	13.0	14.7	14.4
		64	500	0.04	18.1	17.2	16.1	19.5	18.3	17.8
		64	500	0.05	21.8	21.5	18.1	22.4	21.6	21.1
		64	500	0.06	23.23	25.79	23.76	20.53	24.31	23.52
		64	500	0.07	26.40	25.35	25.95	24.79	30.67	26.64
		64	500	0.08	25.9	28.3	32.3	31.5	29.4	29.5
		64	500	0.09	35.1	34.9	35.1	31.6	30.7	33.5
		64	500	0.1	35.38	41.23	35.89	36.58	34.69	36.76

The graph in figure 3 below plots the average round trip transfer time against loss probability. In this graph, values of loss probability are plotted on the x-axis and values obtained for average round trip time are plotted on y-axis.

Project 2: Simple FTP with Go-back-N

Aparna Patil - akpatil
Nikunj Shah - nmshah5

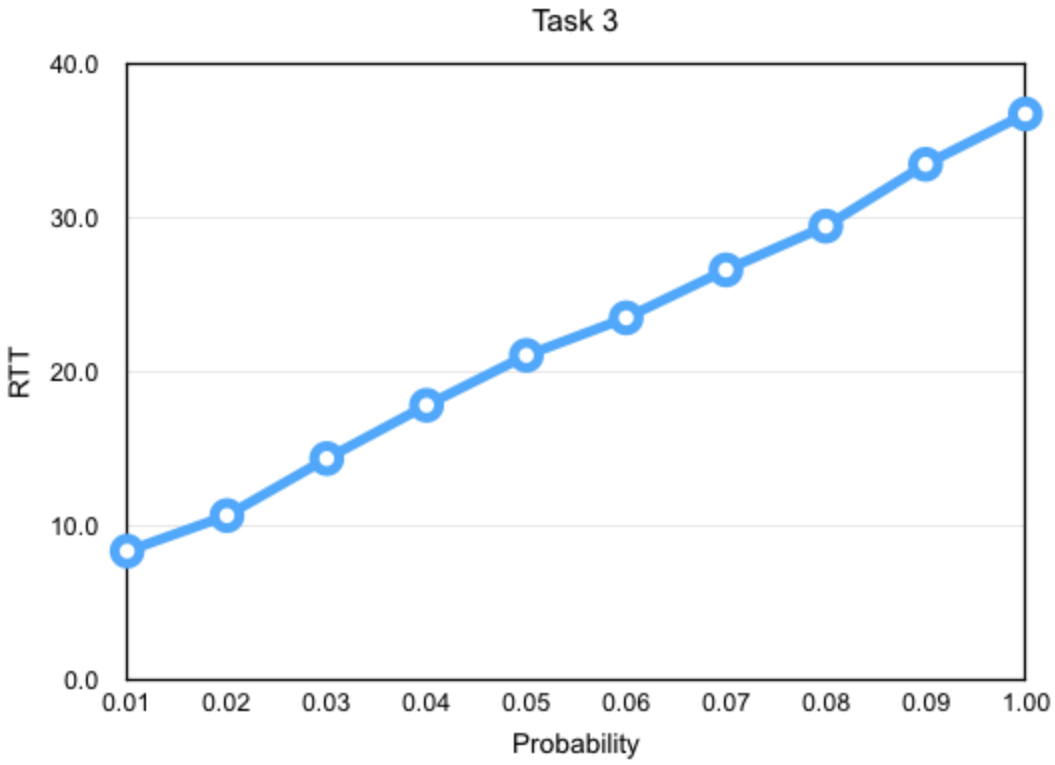

Figure 3: Transfer Time (seconds) vs Probability

Transfer time increases with the increase in loss probability. In Go-back-N protocol as packets are lost at receiver, sender has to resend the packets again until an ACK is received. If the packet error rate increases, then sender has to send packets again hence the time needed to retransmit the lost packets also increases.